Duet Robust Deep Subspace Clustering

Yangbangyan Jiang[1,2], Qianqian Xu[3], Zhiyong Yang[1,2],
Xiaochun Cao[1,2], Qingming Huang[3,4,5*]
[1]State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China
[4]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
[5]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, China
{jiangyangbangyan,yangzhiyong,caoxiaochun}@iie.ac.cn,xuqianqian@ict.ac.cn,qmhuang@ucas.ac.cn

ABSTRACT

Subspace clustering has long been recognized as vulnerable toward gross corruptions – the corruptions can easily mislead the estimation of the underlying subspace structure. Recently, deep extensions of traditional subspace clustering methods have shown their great power to boost the clustering performance. However, deep learning methods are, in themselves, more prone to be affected by data corruptions. This motivates us to design specific robust extensions for deep subspace clustering methods. More precisely, we contribute a new robust deep framework called Duet Robust Deep Subspace Clustering (DRDSC). Our main idea is to explicitly model the corrupted patterns from both the data reconstruction perspective and the latent self-expression perspective with two regularization norms. Moreover, since the two involved norms are non-smooth, we implement a smoothing technique for these norms to facilitate the back-propagation of our proposed network. Experiments carried out on read-world vision tasks with different noise settings demonstrate the effectiveness of our proposed method.

KEYWORDS

subspace clustering, deep learning, robustness

ACM Reference Format:
Yangbangyan Jiang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, Qingming Huang. 2019. Duet Robust Deep Subspace Clustering. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350852

*Corresponding author.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350852

1 INTRODUCTION

Subspace clustering is the process of assigning subspace memberships to a set of unlabeled data points which are roughly drawn from the union of an unknown number of low-dimensional subspaces. During the past decades, subspace clustering has successfully attracted enduring attention from a wide spectrum of multimedia applications, such as metric learning [41], motion segmentation [44], image segmentation [14], and video clustering [23].

As a dominant approach in subspace clustering, spectral type methods represents each sample by a linear combination of the remaining samples and recovers the subspace structure through different structure priors. The early trials in this area restrict the data with a linear subspace assumption [7, 20, 25]. Recently, with the rise of deep learning techniques, deep neural networks, especially deep auto-encoders, are successfully introduced into this problem to tackle non-linear data [15]. By virtue of the powerful unsupervised representation ability, deep learning based subspace clustering has achieved a great triumph against their shallow counterparts.

The aforementioned achievements are mainly observed on reasonable datasets where almost no corruption occurs. However, due to corrupted or irrelevant measurements, gross errors are ubiquitous in real-world data. Unfortunately, subspace clustering is known to be vulnerable to such corrupted observations, even a single grossly corrupted item is sufficient to drive the subspace structure far away from the ground truth. What's more, the extremely high model complexity makes the deep subspace clustering method even more prone to be affected by such adverse corruptions.

Seeing the weakness, a few initial studies toward robust deep subspace clustering have come out as a remedy. Nonetheless, the major issue is that they do not directly tackle the corruptions in data. As a representative example, [16] introduces a robust learning scheme into the deep subspace clustering framework. By dynamically weighting each sample, this approach learns from easy samples to hard ones, and thus prevents the algorithm from getting stuck into bad local minima. However, when the dataset is full of gross errors, the model has to utilize hard examples at the beginning of learning, which puts its validity at risk. Different from these studies, our goal in this paper is to propose a robust deep subspace clustering method where the corruption patterns are specifically modeled based on the structure of the model.

Note that a common deep subspace clustering framework involves two main procedures: (1) transforms non-linear data into a linear latent space and reconstructs the data, (2) represents the latent codes by themselves to find the underlying subspace structure. Apparently, the corrupted instances in data might destroy the model from both data reconstruction and latent self-representation. Worse, the corruption during reconstruction will spread to self-representation through those latent codes with low fidelity. Therefore, there is a necessity to consider the corruption in both two procedures.

In this paper, we propose a novel framework, *duet robust deep subspace clustering* (DRDSC), to handle contaminated data and enhance the robustness. The robustness of our method comes from the following two aspects. First, we propose a robust deep auto-encoder by isolating the corruptions in the input to obtain more faithful reconstruction. Based on the robust backbone, we then further split the possible corrupted errors from clean latent codes to ensure the subspace structure not to be distorted in self-representation. Both corruption terms are penalized by group-sparsity norms. Before optimization, we adopt a smoothing technique to alleviate the non-differentiability of these norms, such that all the variables in the network can be updated by back-propagation. Our experiments on three widely used face/object datasets validate that our approach successfully reduces the distortion from gross errors.

The contributions of our paper are two-fold:

- We propose a novel robust deep subspace clustering framework. By explicitly modeling the corruptions of data construction and latent self-expression respectively, the robustness of our model is preserved in a double-assurance manner.
- We utilize a smoothing technique for norms to yield a differentiable network. The non-differentiability of norms are eliminated through the smooth approximation, such that back-propagation is available to update both the network parameters and the corruption terms.

2 RELATED WORK

Existing subspace clustering methods can be roughly classified into four categories: iterative [13, 49], algebraic [9, 27], statistical [10, 31], and spectral type methods [7, 20]. Recently, spectral type models have attracted a lot of attention. Such algorithms exploit the self-expressiveness of subspaces, where a data point can be represented by a linear combination of other data points in the same subspace. In practice, this type of methods can be decomposed into two steps: (1) build an affinity matrix by estimating the similarity between data pairs based on self-expression; (2) perform spectral clustering on the affinity matrix to obtain the partitions. Obviously, building a suitable affinity matrix is of much significance in these steps. Ideally, the affinity matrix is block diagonal since all the inter-cluster affinities should be zero. To approximate such a structure, sparse or low-rank priors are adopted to the self-expressive coefficient matrix. Typically, Sparse Subspace Clustering (SSC) [7] enforces the coefficients to be sparse, while Low-Rank Representation (LRR) [20] requires this coefficient matrix to be low-rank. Based on these two algorithms, a lot of improvements are made in terms of the effectiveness [19, 28, 46] and efficiency [33, 47]. Unlike these indirect structure priors, Block Diagonal Regularizer (BDR) is proposed in [25] directly pursuing the block diagonal structure by explicitly enforcing the graph Laplacian to be k-connected.

Nevertheless, the above methods are limited to represent the data points with linear combinations, thus might not be effective to handle data sampled from non-linear subspaces. To overcome this disadvantage, existing linear models like SSC and LRR are kernelized for non-linearity [29, 42, 45]. Recently, deep learning based methods have been proposed as a solution to this problem. [15] integrates a self-expressive layer into deep auto-encoder to simultaneously learn the latent non-linear representations and combination coefficients, which considerably improves the clustering performance. Base on this framework, other learning schemes [16, 50] are introduced to jointly guide the sample representation and subspace self-expression.

Robust Subspace Clustering. With the rise of subspace clustering, the corruption existing in data raises the concern of the community. A naive thought is to separate errors from contaminated data [8, 24]. For instance, SSC and LRR both have robust extensions by writing a contaminated element as a linear combination of other elements plus an error [6, 21]. Different from them, Low Rank Subspace Clustering [39] first gets rid of the corruptions in data, then uses clean data to represent themselves. Based on these explicit error models, more techniques are introduced to further improve the robustness, such as dimension reduction [35], thresholding [12, 30], and matching pursuits [37]. Besides, a robust metric like correntropy [11, 26] could help to suppress the influence of corruptions. Meanwhile, a few studies are conducted to analyze subspace clustering methods under noise theoretically [35, 43].

On the other hand, some outlier detection algorithms for subspace clustering [22, 34, 36, 48] have been developed recently. These algorithms aim at detecting and pruning outliers in the dataset, then using only inliers for clustering. In this paper, we focus on subspace clustering with outliers rather than removing them.

In summary, though deep learning based models have achieved promising performance, they still suffer from the gross errors. To address this issue, we present a solution that tries to alleviate the corrupted errors in deep models.

3 METHODOLOGY

In this section, we present our deep robust subspace clustering method. First, we introduce the general formulation of the subspace clustering problem. From this formulation, we easily point out the deficiency of existing robust methods and then move to our proposed model. After detailing the architecture and corresponding robustness of our model, we tackle the

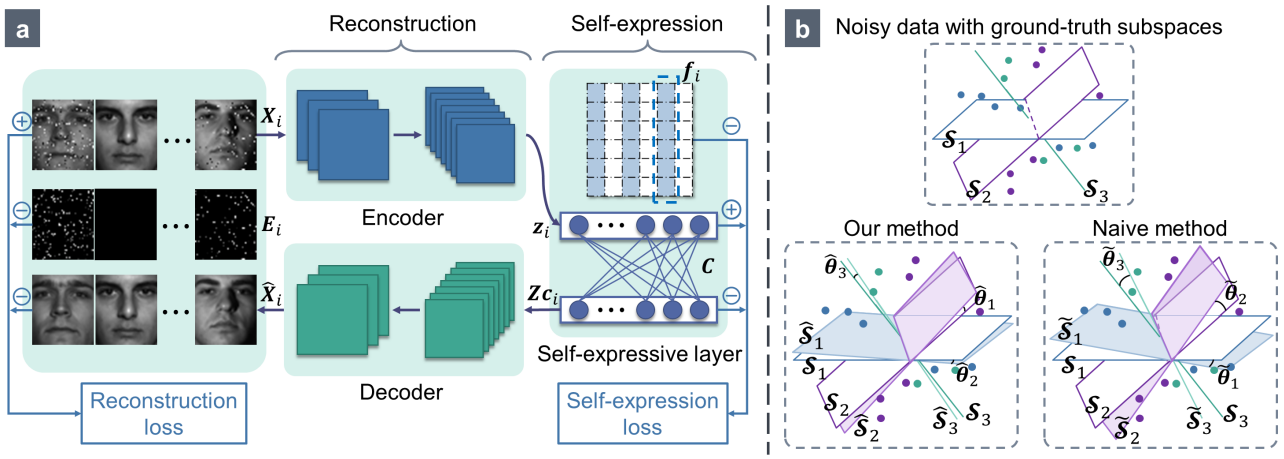


Figure 1: Workflow of Duet Robust Deep Subspace Clustering. (a) The inputted samples $\boldsymbol{X}_i$ contain potential noises. Our framework consists of two main procedures: data reconstruction by an auto-encoder and self-representation by a self-expression module. We consider corruptions coming from both components (with $\boldsymbol{E}_i$ and $\boldsymbol{F}$ as the learned indicators for corresponding corruptions). Our model is optimized under the joint guidance of the Reconstruction loss and Self-expression loss. (b) Different with the existing deep subspace clustering framework, we show our proposed method could significantly prevent the model from being affected by the data corruptions. Namely, subspaces estimated by our method could be closer to the ground-truth than the naive methods.

optimization for the non-differentiable objective function with the smoothing of norms.

3.1 Problem definition

Given a dataset $\mathcal{D} = \{\boldsymbol{X}_i\}_{i=1}^n$ where data points are drawn from a union of k unknown low-dimensional subspaces, our goal is to cluster them into k partitions/subspaces. For the sake of clarity, we limit our discussion to the clustering of images, thus $\boldsymbol{X}_i$ denotes i-th image in the following description.

The most crucial property of subspace clustering is the *self-expressiveness* of subspaces, where all data points can be represented using linear combinations of other points. Obviously, this property holds only when the subspaces are all linear or affine. To cope with data lying in non-linear subspaces, we adopt a series of non-linear transformations to map these data into a latent linear space, *i.e.*, each image $\boldsymbol{X}_i$ is encoded to a latent vector $\boldsymbol{z}_i \in \mathbb{R}^d$. Let $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n] \in \mathbb{R}^{d\times n}$ be the latent code matrix and $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ be the coefficients of the linear combinations. We can formulate the property of self-expressiveness as $\boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{C}$. It is assumed that only a non-trivial c_{ij} indicates that i- and j-th sample lie in the same subspace. Hence, $\boldsymbol{C}$ is expected to be a k block diagonal matrix after properly reordering the data points. Namely, the number of blocks is the number of subspaces. Meanwhile, we also expect all the diagonal entries of $\boldsymbol{C}$ to be zero, such that each point will not be represented by itself.

To obtain compact and effective latent representations, we reconstruct $\boldsymbol{z}_i$ to $\hat{\boldsymbol{X}}_i$ by non-linear transformations and minimize the difference between $\boldsymbol{X}_i$ and $\hat{\boldsymbol{X}}_i$. By converting the self-expressiveness constraint into a penalty term, subspace clustering can be formulated as a minimization problem with a general form as follows:

$$\begin{aligned} \min_{\boldsymbol{C}} \ & \mathcal{L}(\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{C}) + \lambda_1 \sum_i \mathcal{P}(\boldsymbol{X}_i, \hat{\boldsymbol{X}}_i) + \lambda_2 \mathcal{R}(\boldsymbol{C}) \\ s.t. \ & diag(\boldsymbol{C}) = 0 \end{aligned} \tag{1}$$

Here, we use $\mathcal{L}(\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{C})$ to measure the self-expression error, $\mathcal{P}(\boldsymbol{X}_i, \hat{\boldsymbol{X}}_i)$ to calculate the reconstruction cost, and $\mathcal{R}(\boldsymbol{C})$ as a prior structure regularization for the self-expressive coefficients $\boldsymbol{C}$. To approximate a block diagonal $\boldsymbol{C}$, there are three major types of $\mathcal{R}(\boldsymbol{C})$: ℓ_1 norm to pursue sparsity, nuclear norm to pursue low-rankness, and a block diagonal regularizer directly pursuing the block diagonality, which are proposed in SSC, LRR and BDR, respectively. Besides, ℓ_2 norm is also proved to be equivalent to the nuclear norm to obtain a low-rank structure [24].

In the presence of corruptions, most of the existing researches are limited to reduce the influence on $\mathcal{L}(\cdot)$ that could directly distort the estimated subspace structures. Unfortunately, none of them realizes that a corrupted $\boldsymbol{X}$ will induce an unfaithful $\boldsymbol{Z}$ according to $\mathcal{P}(\cdot)$, which leads to an inaccurate self-expression in $\mathcal{L}(\cdot)$ and consequently a sub-optimal solution. Especially when $\mathcal{P}(\cdot)$ is modeled by a deep neural network, the high complexity of deep network will increase the risk of distortion. To address this issue, we propose the *duet robust deep subspace clustering* (DRDSC) model. In this approach, we take both $\mathcal{L}(\cdot)$ and $\mathcal{P}(\cdot)$ into consideration, and propose to split the corruptions from the clean data progressively.

3.2 A Duet Robust Deep Subspace Clustering Framework

As illustrated in Figure 1, our model consists of a deep convolutional auto-encoder as the backbone network and a fully-connected (FC) layer as the self-expressive layer. The model learns latent representations of data by the auto-encoder and conducts self-expression in the latent space through the FC layer. In the following, we show how these two modules enhance the robustness.

Robust Reconstruction with $\mathcal{P}(\cdot)$. The deep convolutional auto-encoder non-linearly transforms an input image $\boldsymbol{X}_i$ into the latent representation $\boldsymbol{z}_i$ via the encoder and then reconstruct the data as $\hat{\boldsymbol{X}}_i$ via the decoder. A common choice for $\mathcal{P}(\cdot)$ is $\frac{1}{2}\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i\|_F^2$. However, such squared term makes this function easily dominated by samples, usually corrupted ones, with large difference between the input and output. To overcome this difficulty, we explicitly model the corruptions of the reconstruction using a set of corruption indicator matrices $\boldsymbol{E}_i$ for each image. Furthermore, we impose the Frobenius-norm on the corruption so that the network would not be dominated by them and can take the best advantage of its representation ability to recover the data. Accordingly, our final $\mathcal{P}(\cdot)$ is formulated as follows:

$$\mathcal{P}(\boldsymbol{X}_i, \hat{\boldsymbol{X}}_i) = \frac{1}{2}\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i - \boldsymbol{E}_i\|_F^2 + \eta_1\|\boldsymbol{E}_i\|_F \qquad (2)$$

where the hyperparameter η_1 controls the tradeoff between reconstruction accuracy and corruption magnitude.

Robust Self-expressiveness with $\mathcal{L}(\cdot)$. After sufficient non-linear transformations, we could assume that the latent representations lie in linear subspaces. Then we employ an FC layer without bias and activation function between the encoder and decoder to represent latent codes with themselves. Accordingly, the parameter of this layer can be modeled as the representation coefficient matrix $\boldsymbol{C}$. Traditional self-expressive loss is usually modeled by:

$$\mathcal{L}(\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{C}) = \frac{1}{2}\|\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{C}\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\|\boldsymbol{z}_i - \boldsymbol{Z}\boldsymbol{c}_i\|_2^2 \qquad (3)$$

where $\boldsymbol{c}_i$ denotes the i-th column of $\boldsymbol{C}$.

Likewise, to prevent the loss from being dominated by the outliers, we explicitly split the corruptions $\boldsymbol{f}_i$ from each latent code $\boldsymbol{z}_i$. Then the robust $\mathcal{L}(\cdot)$ is:

$$\mathcal{L}(\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{C}) = \sum_{i=1}^{n}\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{Z}\boldsymbol{c}_i - \boldsymbol{f}_i\|_2^2 + \eta_2\|\boldsymbol{f}_i\|_2 \qquad (4)$$

where η_2 is a hyperparameter balancing the tradeoff between the two terms. Besides, to approximate a block diagonal $\boldsymbol{C}$, it is common to use the ℓ_p norm as the structure prior constraint $\mathcal{R}(\boldsymbol{C}) = \|\boldsymbol{C}\|_p$.

Putting all these together, we eventually obtain the objective function as follows:

$$\begin{aligned} L = &\sum_{i=1}^{n}\frac{1}{2}\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i - \boldsymbol{E}_i\|_F^2 + \eta_1\|\boldsymbol{E}_i\|_F + \\ &\lambda_1\Big(\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{Z}\boldsymbol{c}_i - \boldsymbol{f}_i\|_2^2 + \eta_2\|\boldsymbol{f}_i\|_2\Big) + \lambda_2\|\boldsymbol{C}\|_p \end{aligned} \qquad (5)$$

Different with Eq. (2), we apply the tradeoff hyperparameter λ_1 to $\mathcal{L}(\cdot)$ instead of $\mathcal{P}(\cdot)$, in respect that the latent space is linear only when the performance of reconstruction is guaranteed.

3.3 Smoothing of norms

In Eq.(5), the Frobenius and ℓ_2 norm of corruption terms are replenished into the network for a double assurance of robustness. However, they also bring the difficulty using gradient-based optimizers due to their non-differentiability at zero. A straightforward solution to this problem is to find smooth approximations of the norms, which are differentiable everywhere while approximates the original value as closely as possible. Therefore, we then introduce a smoothing technique of norms and provide two propositions for practical smooth approximation of Frobenius and ℓ_2 norms.

At first, we give the definition of L-smoothness.

Definition 1 (L-smoothness, [3]). *Let $L \geq 0$. A function $f : \mathbb{E} \to (-\infty, \infty)$ is said to be **L-smooth** over a set $D \subseteq \mathbb{E}$ if it is differentiable over D and satisfies $\sup_{\boldsymbol{x}} \|\nabla^2 f(\boldsymbol{x})\|_{2,2} \leq L$, where $\|\boldsymbol{A}\|_{2,2} = \sqrt{\sum_i \|\boldsymbol{a}_i\|_2}$, and the constant L is called the smoothness parameter.*

By this definition, we have the definition of a smooth approximation of a convex function.

Definition 2 ($\frac{1}{\mu}$-smooth approximation, [2]). *For a convex function $h : \mathbb{E} \to \mathbb{R}$, a convex differentiable function $h_\mu : \mathbb{E} \to \mathbb{R}$ is said to be its $\frac{1}{\mu}$-**smooth approximation** with parameters (α, β) if for any $\mu > 0$ the following holds:*

1) *$h_\mu(\boldsymbol{x}) \leq h(\boldsymbol{x}) \leq h_\mu(\boldsymbol{x}) + \beta\mu$ for all $\boldsymbol{x} \in \mathbb{E}$,*
2) *h_μ is $\frac{\alpha}{\mu}$-smooth.*

Then we have the following proposition to smooth the Frobenius norm.

Proposition 1 (smoothed Frobenius norm). *Define $\|\boldsymbol{X}\|_{F_{(\mu)}} = \sqrt{\|\boldsymbol{X}\|_F^2 + \mu^2} - \mu$, then $\|\cdot\|_{F_{(\mu)}}$ is a $\frac{1}{\mu}$-smooth approximation of the Frobenius norm $\|\cdot\|_F$ with parameters (1,1).*

Similarly, the ℓ_2 norm can be smoothed as follows.

Proposition 2 (smoothed ℓ_2 norm). *Define $\|\boldsymbol{X}\|_{2_{(\mu)}} = \sqrt{\|\boldsymbol{x}\|_2^2 + \mu^2} - \mu$, then $\|\cdot\|_{2_{(\mu)}}$ is a $\frac{1}{\mu}$-smooth approximation of the ℓ_2 norm $\|\cdot\|_2$ with parameters (1,1).*

The smooth approximation provides us with a powerful tool to eliminate the non-differentiability of norms. By choosing different μ, we can balance the approximation accuracy and smoothness. In practice, a very small μ (say 10^{-6} or 10^{-8}) could induce a differentiable approximation with hardly any performance degradation. Thus we fix its value to 10^{-6} in our implementation. Due to the limited space, we present the proof of these two propositions and the sensitivity analysis of μ in supplementary materials.

3.4 Optimization

By Proposition 1 and 2, we obtain two tractable smoothed norms and can rewrite the overall objective as follows:

$$L = \sum_{i=1}^{n} \frac{1}{2}\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i - \boldsymbol{E}_i\|_F^2 + \eta_1 \|\boldsymbol{E}_i\|_{F_{(\mu)}} + \lambda_1 \Big(\frac{1}{2}\|\boldsymbol{z}_i - \boldsymbol{Z}\boldsymbol{c}_i - \boldsymbol{f}_i\|_2^2 + \eta_2 \|\boldsymbol{f}_i\|_{2_{(\mu)}}\Big) + \lambda_2 \|\boldsymbol{C}\|_p \tag{6}$$

Since all terms in Eq. (6) are differentiable, the smoothing of norms yields an totally differentiable network, which can be updated via gradient-based solver like Adam [17]. Namely, all the variables, including the parameters of encoder/decoder, the self-expressive coefficients $\boldsymbol{C}$, and the corruption terms $\{\boldsymbol{E}_i\}$ and $\{\boldsymbol{f}_i\}$, will be simultaneously updated in our model.

When the training finished, we build the symmetric and non-negative affinity matrix by $\boldsymbol{W} = |\boldsymbol{C}| + |\boldsymbol{C}^\top|$ and find the partition through spectral clustering [40] on $\boldsymbol{W}$. Let $\boldsymbol{L}_W = \boldsymbol{D}_W - \boldsymbol{W}$ be the Laplacian matrix and $\boldsymbol{D}_W$ be the degree matrix which is diagonal with the diagonal element as $d_{ii} = \sum_{j=1}^{n} w_{ij}$. We first perform eigen-decomposition on normalized Laplacian $\tilde{\boldsymbol{L}}_W = \boldsymbol{D}_W^{-1/2} \boldsymbol{L}_W \boldsymbol{D}_W^{-1/2}$, then obtain final clustering label $\hat{\boldsymbol{y}}$ by feeding the eigenvectors associated with the bottom k eigenvalues into k-means [1].

4 EXPERIMENTS

In this section, we carry out evaluations on three popular benchmark datasets in different corruption settings.

4.1 Experimental setup

Competitors. We compare our model with the following 10 methods.

(1) **SSC** [7]: The self-expressive coefficient matrix is assumed to be sparse and constrained by ℓ_1 norm.

(2) **LRR** [20]: The coefficient matrix is assumed to be low-rank and constrained by nuclear norm.

(3) **LRSC** [39]: Data are decomposed into clean and corrupted parts and self-represented by only clean data under the low-rank assumption.

(4) **BDR** [25]: An explicit block diagonal regularizer is employed to pursue a block diagonal structure for $\boldsymbol{C}$.

(5) **KSSC** [29]: SSC is combined with the kernel trick to deal with non-linear data.

(6-8) **AE+SSC/LRR/BDR**: The latent representations extracted by a convolutional auto-encoder is used as the input of SSC/LRR/BDR.

(9) **DSC-Net** [15]: An auto-encoder inserted with a self-expressive layer is adopted to jointly reconstruct the data and express the latent codes by themselves.

(10) **DeepCogSC** [16]: An adaptive weighting and easy-to-hard learning scheme is introduced into DSC-Net.

To better show the effectiveness of our proposed method, we implement SSC and LRR by explicitly modeling the self-expressive corruptions with $\ell_{2,1}$ norm to ensure their robustness. Besides, for fair comparison, we adopt the architecture of convolutional auto-encoders in [15] for feature extraction and the backbone network in DSC-Net/DeepCogSC/DRDSC.

(a) (b) (c) (d) (e)

Figure 2: Example of corrupted samples on Extended Yale B dataset. (a) Original image, (b/c) 10%/40% corrupted by random pixel corruption, (d/e) 10%/40% corrupted by Gaussian noise.

And the structure prior in these deep methods is limited to the ℓ_2 norm for clarity. As discussed in Section 3.3, we fix μ value at 10^{-6} in our implementation.

Evaluation metrics. We use 7 metrics to evaluate the clustering performance: Clustering accuracy (**ACC**), Normalized Mutual Information (**NMI**), Precision (**P**), Recall (**R**), F1-score (**F**), Adjusted Rand Index (**ARI**), Purity (**PUR**).

Synthetic corruptions. To explicitly evaluate the performance under different situations, all samples in the dataset are contaminated by two typical kinds of additive corruptions: random pixel corruption and Gaussian noise. For the former, we directly replace randomly selected pixels with values under a uniform distribution on $[0, 255]$, thus the information in those positions is totally lost. Yet the latter noise, with a mean of 0 and a std of 255, is added to the pixel value, such that a part of information is still preserved in those corrupted pixels. The corruption ratio, *i.e.*, the percentage of randomly selected pixels, is set to 10% and 40% to simulate a light and heavy contamination, respectively. Note that the ratio is lower than 50% to guarantee that the model would not confuse the corruptions with clean pixels. Some corrupted examples are depicted in Figure 2.

4.2 Extended Yale B Dataset

Dataset description. First, we evaluate our method on Extended Yale B dataset [18] (EYaleB for short) for the face clustering task. As a popular benchmark, it contains 2432 grayscale face images, taken from 38 subjects under 64 illumination conditions.

Implementation details. The images are down-sampled to the size of 48×42 for preprocessing following the experimental setup in [7]. In training, the learning rate is set to 8×10^{-4}. We set λ_1 to 1, λ_2 to $10^{0.8}$ as in DSC-Net for fair comparison, and set η_1 and η_2 to 50 empirically.

Comparative results. All the evaluation results are depicted in Table 1. The highest value is marked in "magenta" and the second highest is marked in "cyan" to evidently denote the improvement. It is easy to see that our DRDSC model outperforms the baselines in most metrics under different ratios of corruptions. Especially under 10% random pixel corruption, the results of DRDSC is higher than the second best DSC-Net with 3.95%, 20.05%, 1.61%, 19.33%, 27.92%, 5.73%, 3.91% in terms of ACC, AR, NMI, F, P, R, PUR, respectively. Specifically, we have the following observations. (1) Under 10% corruptions, SSC and BDR with auto-encoder features both achieve much better clustering results than using raw input, which benefit from the powerful

Table 1: Performance (%) on Extended Yale B with different corruption types and ratios. The larger, the better. The value marked in "magenta" holds the highest value, and "cyan" holds the second highest.

(a) 10% random pixel corruption.

Method	ACC	AR	NMI	F	P	R	Purity
SSC	53.66	32.33	32.16	34.35	29.72	40.70	61.31
LRR	56.41	32.84	62.45	34.84	30.31	40.97	60.86
LRSC	54.11	29.30	59.54	31.57	25.53	41.36	61.06
KSSC	36.88	22.34	49.26	24.48	23.03	26.13	47.66
BDR	56.37	27.23	61.31	29.61	23.42	40.25	60.28
AE+SSC	61.55	38.03	66.59	33.90	34.30	47.68	67.31
AE+LRR	41.61	15.74	49.11	18.60	14.20	26.94	46.34
AE+BDR	57.44	30.90	59.05	33.02	27.90	40.45	60.69
DSC-Net	**85.03**	**57.77**	84.09	**59.08**	**49.13**	**74.08**	**85.03**
DeepCogSC	84.30	50.46	**84.59**	52.12	40.29	73.78	84.29
Ours	**88.98**	**77.82**	**86.20**	**78.41**	**77.05**	**79.81**	**88.94**

(b) 40% random pixel corruption.

Method	ACC	AR	NMI	F	P	R	Purity
SSC	28.45	12.55	36.51	15.00	13.92	16.25	35.36
LRR	30.43	**13.12**	33.43	**15.49**	**14.77**	16.28	34.66
LRSC	30.14	11.79	35.56	14.31	12.96	15.98	33.06
KSSC	27.34	12.31	35.60	14.75	13.75	15.91	34.09
BDR	30.76	**13.25**	38.70	**15.74**	**14.22**	17.63	36.23
AE+SSC	27.47	11.65	34.72	14.07	13.33	14.89	31.58
AE+LRR	24.88	11.59	34.61	13.96	13.53	14.41	30.51
AE+BDR	28.78	12.75	36.25	15.20	14.05	16.55	34.54
DSC-Net	30.67	8.52	35.53	11.64	8.85	16.97	35.52
DeepCogSC	**32.15**	4.88	**42.59**	9.03	5.37	**28.53**	**36.88**
Ours	**31.58**	3.35	**45.85**	7.91	4.34	**44.39**	**37.09**

(c) 10% Gaussian noise.

Method	ACC	AR	NMI	F	P	R	Purity
SSC	56.04	31.94	61.13	34.00	29.07	40.94	61.68
LRR	56.50	30.47	62.83	32.64	27.07	41.10	60.73
LRSC	53.12	25.16	60.26	27.67	21.41	39.08	58.47
KSSC	37.34	22.59	49.65	24.75	22.99	26.80	48.19
BDR	55.43	23.48	59.61	26.11	19.72	38.60	58.06
AE+SSC	61.47	37.99	66.58	39.86	34.21	47.75	67.15
AE+LRR	41.24	17.06	49.13	19.78	15.61	26.98	47.20
AE+BDR	57.24	29.90	63.23	32.15	25.93	42.29	62.34
DSC-Net	81.70	**44.40**	82.01	**46.32**	**34.63**	69.94	81.70
DeepCogSC	**82.65**	43.91	**93.59**	45.89	33.73	**71.72**	**82.65**
Ours	**89.43**	**79.47**	**86.75**	**80.01**	**79.25**	**80.79**	**89.43**

(d) 40% Gaussian noise.

Method	ACC	AR	NMI	F	P	R	Purity
SSC	30.22	**13.47**	37.67	**15.92**	**14.56**	17.55	36.47
LRR	28.91	12.72	32.57	15.06	**14.56**	15.59	33.10
LRSC	31.04	13.33	37.18	15.82	14.29	17.71	35.44
KSSC	26.32	11.71	33.67	14.13	13.34	15.03	32.85
BDR	32.32	**14.95**	39.71	**17.35**	**15.92**	19.08	38.65
AE+SSC	27.06	11.65	34.72	14.07	13.33	14.89	31.58
AE+LRR	24.47	10.45	33.09	12.86	12.40	13.35	28.82
AE+BDR	28.50	12.43	34.74	14.79	14.21	15.43	32.94
DSC-Net	28.33	8.67	35.04	11.74	9.06	16.66	32.36
DeepCogSC	**34.58**	5.09	**42.40**	9.23	5.48	**29.19**	**38.73**
Ours	**34.91**	5.32	**45.29**	9.48	5.60	**30.93**	**38.94**

feature extraction ability of DNNs. By contrast, the results of AE+LRR are lower than LRR. This is probably because the low-rankness is more fragile to corruptions than sparsity on this dataset. (2) However, under 40% corruptions, SSC, LRR and BDR all outperform AE+SSC/LRR/BDR. Since the auto-encoder is trained with uncontaminated data, such results confirm that contaminated features extracted by DNN can also distort the underlying subspace structure. (3) With the double assurance of robustness in data reconstruction and latent self-representation, our DRDSC further reaches the highest performance under slight corruptions. However, when corruptions are heavy, DRDSC only outperforms the baselines in terms of ACC, NMI, R and PUR. The reason might be that both DRDSC and the second best DeepCogSC focus on the agreement in the clustering results, *i.e.*, the ratio of similar pairs that are assigned to the same cluster. Thus our model is more powerful in terms of these related metrics.

4.3 ORL Dataset

Dataset description. The ORL dataset [32] consists of 400 face images for 40 subjects. Each subject has 10 images taken at different times, lighting conditions and facial expressions. **Implementation details.** Similarly, following [5], the images are down-sampled to the size of 32×32 for preprocessing. We set the learning rate to 5.5×10^{-4} for training. Then we fix λ_1 at 1.0, λ_2 at 0.2 as DSC-Net, while choosing η_1 and η_2 at 250 experimentally.

Comparative results. The quantitative results are summarized in Table 2. Our DRDSC framework achieves better performance than other competitors in all the metrics under 10%/40% random pixel corruption and 40% Gaussian noise. It is worth mentioning that, *with regard to ACC, AR, NMI, F, P, R and PUR, our model outperforms the second best method by up to 2.5%, 2.65%, 1.6%, 2.62%, 2.19%, 3.06%, 1.5% improvement, respectively.* Besides, the following observations can be made. (1) Unlike on EYaleB, our DRDSC could still perform superior to other methods in terms of all metrics under heavy corruptions. The reason might be that images in ORL are easier to cluster than EYaleB even in the presence of gross errors. (2) BDR and AE+BDR both beat other non-deep models, and AE+BDR even obtains comparable results to deep models. This indicates that a direct block diagonal regularizer might be more effective than regularization using indirect structure priors in self-expression.

4.4 COIL20 Dataset

Dataset description. Then we evaluate the methods on a more general task, object clustering. The COIL20 dataset [32] includes 1440 images for 20 objects, each of which has 72 images taken at different pose intervals on a turntable. Unlike previous datasets, there are many blank pixels in the images. Hence extra corruptions are prone to distort the latent patterns, which makes this dataset hard to cluster.

Table 2: Performance (%) on ORL with different corruption types and ratios.

(a) 10% random pixel corruption.

Method	ACC	AR	NMI	F	P	R	PUR
SSC	69.25	57.35	82.59	58.34	56.68	60.11	72.75
LRR	70.25	58.73	82.57	59.71	57.06	62.61	74.25
LRSC	73.50	61.19	84.62	62.12	58.49	66.22	76.50
KSSC	73.75	61.47	84.41	62.38	59.51	65.56	76.00
BDR	75.00	63.73	85.99	64.59	62.15	67.22	77.75
AE+SSC	71.00	57.40	82.04	58.41	55.35	61.83	73.00
AE+LRR	69.50	58.74	83.08	59.71	57.34	62.28	72.50
AE+BDR	75.25	65.99	87.41	66.79	63.68	70.22	79.00
DSC-Net	**81.50**	**71.57**	88.02	**72.22**	**71.30**	73.17	82.50
DeepCogSC	**81.50**	71.33	**88.44**	72.00	69.66	**74.50**	**83.00**
Ours	**84.00**	**73.03**	**88.59**	**73.65**	**72.14**	**75.22**	**83.25**

(b) 40% random pixel corruption.

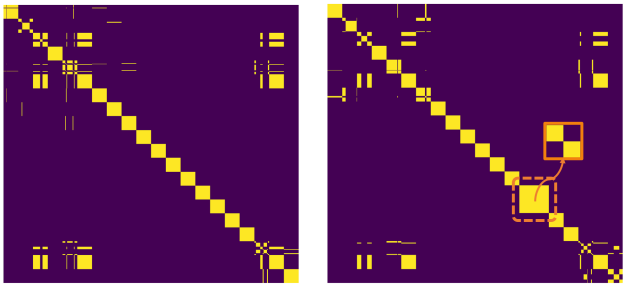
Method	ACC	AR	NMI	F	P	R	PUR
SSC	37.25	16.98	57.01	18.94	18.16	19.78	40.00
LRR	37.00	18.90	59.82	20.82	19.89	21.83	41.75
LRSC	39.75	21.36	60.21	23.40	20.39	27.44	46.25
KSSC	39.00	20.14	58.30	22.02	21.11	23.00	42.25
BDR	39.25	20.24	58.46	22.10	21.37	22.89	41.00
AE+SSC	40.75	19.82	58.94	21.69	20.99	22.44	43.50
AE+LRR	39.75	20.72	60.85	22.59	21.60	23.67	42.50
AE+BDR	45.25	25.27	62.82	27.02	26.10	28.00	48.50
DSC-Net	44.50	25.00	63.20	26.74	26.07	27.44	48.25
DeepCogSC	**47.25**	**26.11**	**63.17**	**27.81**	**27.16**	**28.50**	**49.75**
Ours	**49.75**	**28.75**	**64.80**	**30.42**	**29.35**	**31.56**	**51.50**

(c) 10% Gaussian noise.

Method	ACC	AR	NMI	F	P	R	PUR
SSC	71.75	58.77	83.45	59.73	57.90	61.67	73.25
LRR	68.00	55.96	81.73	56.99	54.92	59.22	71.25
LRSC	69.00	56.79	81.98	57.82	54.86	61.11	72.75
KSSC	71.25	58.62	83.30	59.60	56.69	62.83	73.75
BDR	74.50	63.95	86.25	64.81	61.54	68.44	77.50
AE+SSC	67.25	55.74	82.12	56.78	54.34	59.44	71.00
AE+LRR	71.50	58.42	83.03	59.39	56.86	62.17	74.00
AE+BDR	79.00	71.24	89.44	71.92	68.67	75.50	83.50
DSC-Net	**82.25**	**74.35**	**90.20**	**74.95**	**72.17**	**77.94**	**83.75**
DeepCogSC	81.75	73.46	86.69	74.08	72.00	76.28	83.25
Ours	**83.75**	**75.49**	**89.75**	**76.05**	**74.71**	**77.44**	**85.25**

(d) 40% Gaussian noise.

Method	ACC	AR	NMI	F	P	R	PUR
SSC	39.50	18.57	58.61	20.49	19.58	21.50	41.75
LRR	40.25	21.30	60.37	23.14	22.30	24.06	43.00
LRSC	39.00	19.85	59.00	21.83	20.00	24.00	42.75
KSSC	39.25	18.91	58.66	20.85	19.64	22.22	41.75
BDR	37.00	22.64	61.29	24.63	21.56	**28.72**	45.25
AE+SSC	40.75	22.06	62.17	23.88	23.01	24.83	43.25
AE+LRR	39.25	21.33	60.90	23.18	22.31	24.11	42.50
AE+BDR	43.75	24.27	**63.08**	25.52	25.24	25.56	**47.75**
DSC-Net	**44.00**	24.11	62.93	25.89	24.91	26.94	47.5
DeepCogSC	43.25	**24.78**	62.84	**26.53**	**25.73**	27.39	45.75
Ours	**45.50**	**27.33**	**63.68**	**29.05**	**27.77**	**30.44**	**48.00**



(a) DRDSC **(b) DSC-Net**

Figure 3: Visualization of indicator matrix $\boldsymbol{H}$ on COIL20 under 10% random pixel corruptions. DSC-Net completely confuses two clusters and merge them as one, while our DRDSC could still preserve the correct structure.

Implementation details. The images are down-sampled to 32×32 according to [4]. The learning rate is set to 8.5×10^{-4}. And we fix λ_1 at 1.0 and λ_2 at 150 same as DSC-Net, then set η_1 and η_2 at 10 experimentally.

Comparative results. The clustering results are demonstrated in Table 3. Similar to ORL, DRDSC consistently achieves the highest performance regarding most of the metrics. However, for NMI and R, DeepCogSC is more effective than our model because of its easy-to-hard learning scheme.

To better exhibit the strength of our model, we further visualize the indicator matrix $\boldsymbol{H}$ for the results of DRDSC and DSC-Net under 10% random pixel corruption in Figure 3. Here $\boldsymbol{H}$ is obtained by $h_{ij} = \mathbf{1}[\hat{\boldsymbol{y}}_i = \hat{\boldsymbol{y}}_j]$. As shown in Figure 3, both models obtain a clear block diagonal structure. However, compared with DRDSC, there is an evident error in $\boldsymbol{H}$ of DSC-Net that samples lying in two different subspaces are assigned to the same cluster, which is shown in the orange boxed region. Since DSC-Net does not have the ability to eliminate the influence of contamination, the corruptions easily broadcast to the process of self-expression and thus induce an incorrect subspace structure estimation. This visualization suggests that, our DRDSC effectively prevents the subspace structure from being distorted by the corruptions.

Moreover, we also visualize the spectral embeddings (obtained in spectral clustering) for DRDSC and DeepCogSC using t-SNE [38] under the same setting in Figure 4. Each cluster is marked with a specific color in the figure. In the boxed region of Figure 4b, we can see that DeepCogSC completely mixes up the clusters marked in red and green. In the contrast, our DRDSC could distinctly split these two clusters on the basis of its duet robustness. See Figure 4a. Thus the superiority of our proposed model is again demonstrated. High resolution figures are provided in supplementary materials.

Ablation study. To clarify the contribution for noise modeling on data reconstruction and self-expression, we evaluate two ablated variants on this dataset: 1) DRDSC without robust data reconstruction, 2) DRDSC without robust self-expression (denoted as "w/o-rec" and "w/o-self" respectively). From Table 4, we can observe that one-fold noise modeling

Table 3: Performance (%) on COIL20 with different corruption types and ratios.

(a) 10% random pixel corruption.

Method	ACC	AR	NMI	F	P	R	PUR
SSC	69.93	62.05	79.90	64.05	60.03	68.64	73.06
LRR	69.37	61.50	78.70	63.52	59.73	67.82	77.36
LRSC	68.54	61.58	78.24	63.53	6189	65.25	72.29
KSSC	74.79	71.35	86.31	72.89	66.74	80.29	84.51
BDR	69.44	61.60	79.85	63.63	5949	68.38	74.24
AE+SSC	72.85	67.16	81.86	68.83	66.81	70.98	78.68
AE+LRR	72.71	66.00	80.38	67.70	66.60	68.84	75.90
AE+BDR	71.53	63.70	83.14	65.71	58.42	75.08	82.08
DSC-Net	83.40	**79.59**	**93.13**	**80.72**	**72.06**	**91.74**	85.03
DeepCogSC	**83.47**	78.97	**93.68**	80.15	70.88	**92.21**	**86.32**
Ours	**86.11**	**80.65**	90.96	**81.65**	**78.36**	85.22	**87.08**

(b) 40% random pixel corruption.

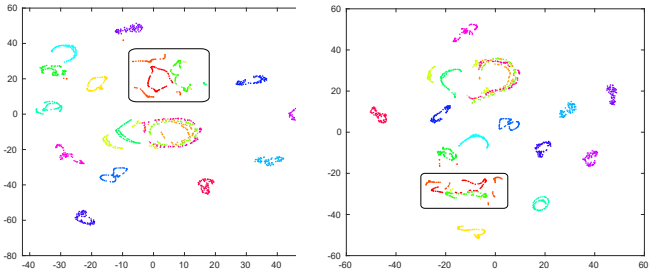
Method	ACC	AR	NMI	F	P	R	PUR
SSC	64.10	54.78	73.43	57.08	55.51	58.73	68.13
LRR	62.36	53.84	71.40	56.15	55.34	56.99	64.93
LRSC	65.49	57.92	74.60	60.05	**58.61**	61.57	70.07
KSSC	67.06	**61.59**	77.17	**61.80**	**58.39**	65.65	72.50
BDR	65.83	57.58	74.93	59.74	57.87	61.75	69.72
AE+SSC	65.56	58.25	75.30	60.42	57.53	63.60	72.22
AE+LRR	63.82	56.27	73.59	58.47	57.53	59.43	67.36
AE+BDR	67.33	**60.26**	77.76	61.03	57.31	65.27	72.22
DSC-Net	**67.57**	56.90	83.01	59.48	49.03	75.61	**73.96**
DeepCogSC	66.60	58.81	**84.57**	61.33	49.47	**80.67**	72.50
Ours	**69.31**	57.74	**84.07**	**61.43**	51.69	**75.69**	**74.86**

(c) 10% Gaussian noise.

Method	ACC	AR	NMI	F	P	R	PUR
SSC	70.21	63.63	79.34	65.47	63.91	67.10	73.33
LRR	67.08	59.24	76.34	61.31	59.53	63.20	72.71
LRSC	68.82	59.44	74.47	61.47	60.60	62.37	70.63
KSSC	73.61	65.42	82.90	67.35	59.77	77.12	81.39
BDR	69.93	60.00	79.15	62.14	57.35	67.79	73.96
AE+SSC	71.60	64.61	79.53	66.38	65.39	67.40	74.51
AE+LRR	71.60	64.61	79.53	66.38	65.39	67.40	74.51
AE+BDR	71.94	65.09	80.41	66.85	65.44	68.32	73.75
DSC-Net	82.22	77.41	92.30	78.67	69.62	90.41	85.76
DeepCogSC	**83.99**	**79.77**	**93.49**	**80.47**	**71.24**	**92.45**	**86.74**
Ours	**84.97**	**79.85**	**93.21**	**81.11**	**72.32**	**91.96**	**87.29**

(d) 40% Gaussian noise.

Method	ACC	AR	NMI	F	P	R	PUR
SSC	65.83	58.69	76.33	60.78	59.35	62.28	69.58
LRR	64.79	56.82	73.84	58.99	57.99	60.02	68.40
LRSC	64.58	56.38	74.56	58.64	55.97	61.58	72.01
KSSC	68.33	61.99	77.52	63.04	**62.21**	63.89	71.11
BDR	66.39	59.77	76.92	61.82	59.84	63.94	71.25
AE+SSC	66.60	56.81	74.21	58.99	57.68	60.35	68.96
AE+LRR	65.76	56.42	74.60	58.64	56.87	60.52	70.07
AE+BDR	68.33	60.17	76.68	62.16	61.50	62.83	70.90
DSC-Net	75.21	66.49	**89.12**	68.46	57.24	**85.16**	80.97
DeepCogSC	**76.60**	**68.75**	**90.15**	**70.59**	59.18	**87.46**	**82.78**
Ours	**79.89**	**70.13**	87.60	**71.78**	**64.19**	81.40	**82.64**



(a) DRDSC **(b) DeepCogSC**

Figure 4: Visualization of spectral embeddings on COIL20 under 10% random pixel corruption. Each cluster is marked in a specific color. In the boxed region, DeepCogSC completely mixes up the red and green clusters, while DRDSC can distinguish them.

Table 4: Ablation study on COIL20.

Noise	Alg.	ACC	ARI	NMI	F	P	R	PUR
Rand10%	w/o-rec	84.24	79.19	92.72	80.35	71.43	91.81	86.32
	w/o-self	84.17	78.64	89.55	79.74	76.27	83.54	85.21
	Ours	**86.11**	**80.65**	**90.96**	**81.65**	**78.36**	**85.22**	**87.08**
Rand40%	w/o-rec	66.32	54.87	83.25	57.74	44.82	81.15	71.11
	w/o-self	69.32	56.28	83.89	59.03	46.35	81.25	71.74
	Ours	**69.31**	**57.74**	**84.07**	**61.43**	**51.69**	**75.69**	**74.86**
Gauss10%	w/o-rec	84.51	77.87	89.79	79.06	72.61	86.77	84.51
	w/o-self	84.31	77.79	90.57	79.00	71.68	87.99	84.51
	Ours	**84.97**	**79.85**	**93.21**	**81.11**	**72.32**	**91.96**	**87.29**
Gauss40%	w/o-rec	71.18	60.75	83.88	63.17	50.54	84.22	71.18
	w/o-self	71.04	63.22	86.21	65.49	52.35	87.43	73.19
	Ours	**79.89**	**70.13**	**87.60**	**71.78**	**64.19**	**81.40**	**82.64**

indeed helps the model address corrupted data. The proposed method still outperforms these variants, which again demonstrates the effectiveness of two-fold noise modeling.

5 CONCLUSION

In this paper, we propose a robust deep subspace clustering framework called *Duet Robust Deep Subspace Clustering* (DRDSC). DRDSC considers the robustness from both data reconstruction and latent self-expression perspective. By explicitly modeling the corruptions in these procedures via two regularization terms, we could minimize the distortion on the underlying subspace structure. Subsequently, a tractable smoothing technique is proposed to overcome the non-smoothness of these norms, thus back-propagation is enabled for the whole network. Finally, comprehensive experiments on three real-world datasets under different corruptions demonstrate the effectiveness of our model.

ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China (No. 61620106009, U1636214, U1803264, 61672514), in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Beijing Natural Science Foundation (No. KZ201910005007, 4182079), and in part by Youth Innovation Promotion Association CAS.

REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *SODA*. 1027–1035.

[2] Amir Beck. 2017. *First-Order Methods in Optimization*. Vol. 25. SIAM.

[3] Sébastien Bubeck et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning* 8, 3-4 (2015), 231–357.

[4] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. 2011. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE TPAMI* 33, 8 (2011), 1548–1560.

[5] Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas S. Huang. 2007. Learning a Spatially Smooth Subspace for Face Recognition. In *CVPR*. 1–7.

[6] Ehsan Elhamifar and René Vidal. 2009. Sparse subspace clustering. In *CVPR*. 2790–2797.

[7] Ehsan Elhamifar and René Vidal. 2013. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE TPAMI* 35, 11 (2013), 2765–2781.

[8] Paolo Favaro, René Vidal, and Avinash Ravichandran. 2011. A closed form solution to robust subspace estimation and clustering. In *CVPR*. 1801–1807.

[9] C. William Gear. 1998. Multibody Grouping from Motion Images. *IJCV* 29, 2 (1998), 133–150.

[10] Amit Gruber and Yair Weiss. 2004. Multibody Factorization with Uncertainty and Missing Data Using the EM Algorithm. In *CVPR*. 707–714.

[11] Ran He, Yingya Zhang, Zhenan Sun, and Qiyue Yin. 2015. Robust Subspace Clustering With Complex Noise. *IEEE TIP* 24, 11 (2015), 4001–4013.

[12] Reinhard Heckel and Helmut Bölcskei. 2015. Robust Subspace Clustering via Thresholding. *IEEE TIT* 61, 11 (2015), 6320–6342.

[13] Jeffrey Ho, Ming-Hsuan Yang, Jongwoo Lim, Kuang-Chih Lee, and David J. Kriegman. 2003. Clustering Appearances of Objects Under Varying Illumination Conditions. In *CVPR*. 11–18.

[14] Sajid Javed, Arif Mahmood, Thierry Bouwmans, and Soon Ki Jung. 2017. Background-Foreground Modeling Based on Spatiotemporal Sparse Subspace Clustering. *IEEE TIP* 26, 12 (2017), 5840–5854.

[15] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian D. Reid. 2017. Deep Subspace Clustering Networks. In *NIPS*. 23–32.

[16] Yangbangyan Jiang, Zhiyong Yang, Qianqian Xu, Xiaochun Cao, and Qingming Huang. 2018. When to Learn What: Deep Cognitive Subspace Clustering. In *ACM MM*. 718–726.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Kuang-Chih Lee, Jeffrey Ho, and David J. Kriegman. 2005. Acquiring Linear Subspaces for Face Recognition under Variable Lighting. *IEEE TPAMI* 27, 5 (2005), 684–698.

[19] Chun-Guang Li, Chong You, and René Vidal. 2017. Structured Sparse Subspace Clustering: A Joint Affinity Learning and Subspace Clustering Framework. *IEEE TIP* 26, 6 (2017), 2988–3001.

[20] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. 2013. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE TPAMI* 35, 1 (2013), 171–184.

[21] Guangcan Liu, Zhouchen Lin, and Yong Yu. 2010. Robust Subspace Segmentation by Low-Rank Representation. In *ICML*. 663–670.

[22] Guangcan Liu, Huan Xu, and Shuicheng Yan. 2012. Exact Subspace Segmentation and Outlier Detection by Low-Rank Representation. In *AISTATS*. 703–711.

[23] Haijun Liu, Jian Cheng, and Feng Wang. 2018. Sequential Subspace Clustering via Temporal Smoothness for Sequential Data Segmentation. *IEEE TIP* 27, 2 (2018), 866–878.

[24] Risheng Liu, Zhouchen Lin, Fernando De la Torre, and Zhixun Su. 2012. Fixed-rank representation for unsupervised visual learning. In *CVPR*. 598–605.

[25] Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, and Shuicheng Yan. 2019. Subspace Clustering by Block Diagonal Representation. *IEEE TPAMI* 41, 2 (2019), 487–501.

[26] Can-Yi Lu, Jinhui Tang, Min Lin, Liang Lin, Shuicheng Yan, and Zhouchen Lin. 2013. Correntropy Induced L2 Graph for Robust Subspace Clustering. In *ICCV*. 1801–1808.

[27] Yi Ma, Allen Y. Yang, Harm Derksen, and Robert M. Fossum. 2008. Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data. *SIAM Rev.* 50, 3 (2008), 413–458.

[28] Vishal M. Patel, Hien Van Nguyen, and René Vidal. 2013. Latent Space Sparse Subspace Clustering. In *ICCV*. 225–232.

[29] Vishal M. Patel and René Vidal. 2014. Kernel sparse subspace clustering. In *ICIP*. 2849–2853.

[30] Xi Peng, Zhang Yi, and Huajin Tang. 2015. Robust Subspace Clustering via Thresholding Ridge Regression. In *AAAI*. 3827–3833.

[31] Shankar R. Rao, Roberto Tron, René Vidal, and Yi Ma. 2010. Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories. *IEEE TPAMI* 32, 10 (2010), 1832–1845.

[32] Ferdinand Samaria and Andy Harter. 1994. Parameterisation of a stochastic model for human face identification. In *WACV*. 138–142.

[33] Jie Shen, Ping Li, and Huan Xu. 2016. Online Low-Rank Subspace Clustering by Basis Dictionary Pursuit. In *ICML*. 622–631.

[34] Mahdi Soltanolkotabi, Emmanuel J Candes, et al. 2012. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics* 40, 4 (2012), 2195–2238.

[35] Mahdi Soltanolkotabi, Ehsan Elhamifar, Emmanuel J Candes, et al. 2014. Robust subspace clustering. *The Annals of Statistics* 42, 2 (2014), 669–699.

[36] Mario Sznaier and Octavia I. Camps. 2018. SoS-RSC: A Sum-of-Squares Polynomial Approach to Robustifying Subspace Clustering Algorithms. In *CVPR*. 8033–8041.

[37] Michael Tschannen and Helmut Bölcskei. 2018. Noisy Subspace Clustering via Matching Pursuits. *IEEE TIT* 64, 6 (2018), 4081–4104.

[38] Laurens van der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *JMLR* 15, 1 (2014), 3221–3245.

[39] René Vidal and Paolo Favaro. 2014. Low rank subspace clustering (LRSC). *Pattern Recognition Letters* 43 (2014), 47–61.

[40] Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17, 4 (2007), 395–416.

[41] Jun Wang, Zhaohong Deng, Kup-Sze Choi, Yizhang Jiang, Xiaoqing Luo, Fu-Lai Chung, and Shitong Wang. 2016. Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition* 52 (2016), 113–134.

[42] Joseph Wang, Venkatesh Saligrama, and David A. Castañón. 2011. Structural similarity and distance in learning. In *Allerton Conference*. 744–751.

[43] Yu-Xiang Wang and Huan Xu. 2016. Noisy Sparse Subspace Clustering. *JMLR* 17, 1 (2016), 320–360.

[44] Guiyu Xia, Huaijiang Sun, Lei Feng, Guoqing Zhang, and Yazhou Liu. 2018. Human Motion Segmentation via Robust Kernel Sparse Subspace Clustering. *IEEE TIP* 27, 1 (2018), 135–150.

[45] Ming Yin, Yi Guo, Junbin Gao, Zhaoshui He, and Shengli Xie. 2016. Kernel Sparse Subspace Clustering on Symmetric Positive Definite Manifolds. In *CVPR*. 5157–5164.

[46] Chong You, Chi Li, Daniel P. Robinson, and René Vidal. 2018. A Scalable Exemplar-Based Subspace Clustering Algorithm for Class-Imbalanced Data. In *ECCV*. 68–85.

[47] Chong You, Daniel P. Robinson, and René Vidal. 2016. Scalable Sparse Subspace Clustering by Orthogonal Matching Pursuit. In *CVPR*. 3918–3927.

[48] Chong You, Daniel P. Robinson, and René Vidal. 2017. Provable Self-Representation Based Outlier Detection in a Union of Subspaces. In *CVPR*. 4323–4332.

[49] Teng Zhang, Arthur Szlam, and Gilad Lerman. 2009. Median k-flats for hybrid linear modeling with many outliers. In *ICCV Workshops*. 234–241.

[50] Pan Zhou, Yunqing Hou, and Jiashi Feng. 2018. Deep Adversarial Subspace Clustering. In *CVPR*. 1596–1604.